

Remarks

This is in response to the Office Action dated 28 March 2006. Applicant respectfully requests reconsideration and allowance of the subject Application.

No Claims are cancelled by this amendment and no Claims are added.

Claims 30 and 37 are amended. Accordingly, Claims 30-37 are pending in this application.

Claim Objections

The amendments to claims 30 and 37 are purely of form to correct informalities noted by the Office, and are not to overcome prior art or any other objections/rejections. In particular, Applicant amended Claims 30 (at line 13) and 37 (at line 12) to correctly recite the phrase "based on at least."

35 U.S.C. § 103 CLAIM REJECTION

Claims 30-37 were rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 6,018,805 to Ma, et al (hereinafter "Ma"), in view of U.S. Patent No. 6,351,487 to Lu (hereinafter "Lu"). Applicant respectfully traverses the rejection.

Claimed Invention

Claims 30-37 are directed to methods of, and systems for, recovering from a failure of server with a client. The method maintains the state of the connection to the client process layer, and not the state of the process layer. A layer of software called "wrapping" surrounds the connection-oriented network protocol layer (such as a TCP layer) and intercepts all communication with that layer, such as communication originating from a network layer or a process layer. The intercepted communications and connection state information associated with intercepted communications received from the wrapping layers are logged.

When it is determined that a connection with the server fails, the wrapping layers respond to the client on behalf of the server based in part, on the logged connection state information. A state of connection associated with the connection-oriented layer prior to the failure is restored, based in part, on the connection state information received from the wrapping layers. There is no need to interpose a proxy or intermediary between the client computer and the server computer. Servers using the invention can crash and recover without dropping connections. The act of restoring the state of connection associated with the connection-oriented layer is invisible to the client.

References

The Office cites Ma & Lu in its § 103 rejection of Claims 30-37.

Ma teaches a recoverable distributed-object application having a client object running on a client machine on a network and a *first* server object on a first server machine on the network with an **intelligent proxy** running on the client machine. Col. 2, lines 24-28. Ma also teaches establishing a connection to a *second* server. Col. 4, lines 40-46. When the client communicates with the server, a proxy for the server is created on the client machine. Col. 4, lines 36-42. The intelligent proxy establishes the new connection to a *second* server when the *first* server object does not respond. Col. 2, lines 40-45. The system of Ma is dependent upon middle-ware layers, skeletons and proxies. Col. 4, lines 10-11.

Ma does **not** teach a method of recovery from a failure of a server to a client that does not need to interpose a proxy or intermediary between the client and server. (See specification, page 18) Likewise, Ma fails to teach responding

to the client on behalf of the server by wrapping layers and without the need of a second server.

Lu is directed to a computer system using a first DSL modem to communicate packets to a second DSL modem after communication is established between the two modems. (Lu, Abstract). Lu is related to DSL technology and is directed to a system with a modem device driver. (Lu, Col. 1, lines 38-40). Lu does not teach wrapping layers as taught by the present claimed invention. Lu specifically teaches NDIS wrapping, which relates to modem drivers and not a process layer. NDIS is a network driver interface specification whose purpose is for communication so that a modem device driver doesn't need to communicate directly to a stack driver. (Lu, Col. 17, lines 55-57). The NDIS wrapper provides a device driver programming interface allowing multiple network protocols to share the same network. (Lu, Col 17, lines 65- Col. 18, lines 1-3).

Claim Analysis for § 103 Rejection:

Independent Claim 30 recites:

A method of recovering from a failure of a server to a client, comprising:

using wrapping layers to intercept communications to a connection-oriented protocol layer, the communications originating from a network layer and a process layer of a layered communications framework, wherein a first wrapping layer is interposed between the process layer and the connection-oriented protocol layer, and wherein a second wrapping layer is interposed between the network layer and the connection-oriented layer;

logging the intercepted communications and connection state information associated with intercepted communications received from the wrapping layers;

determining when a connection with the server fails;

responding to the client on behalf of the server by the wrapping layers based on at least, in part, on the logged connection state information; and

restoring a state of connection associated with the connection-oriented layer prior to the failure, based on at least, in part, on the connection state information received from the wrapping layers, wherein restoring the state of connection associated with the connection-oriented layer is invisible to the client.

Ma fails to teach or suggest the method of Claim 30 because Ma requires the use of *proxies* and requires the use of a *second* server. According to the teachings of Ma, when the client communicates with the server, a *proxy* for the server is created on the client machine. Col. 4, lines 36-42. The intelligent *proxy* establishes the new connection to a *second* server when the *first* server object does not respond. Col. 2, lines 40-45. The system of Ma is dependent upon middle-ware layers, skeletons and proxies. (Col. 4, lines 10-11), whereas the present claimed invention (Claim 30) does **not** need to interpose a proxy or intermediary between the client computer and the server computer.

The Office admits that Ma fails to disclose “proxies being wrappers, and wrappers interposed between the network layer and the connection-oriented layer.” However, the Office contends that Lu teaches wrappers (60, 64) (col. 36-48 and also teaches wrappers being interposed between a network layer (54) and a connection oriented layer (58) (col. 17, lines 38-48, Fig. 5).

Applicant sincerely believes the Office to have mischaracterized the art by purporting that a proxy is the equivalent of a wrapper and that the two are interchangeable. The Office states at paragraph 7 that “Examiner has interpreted the proxies taught by Ma to be the wrappers in the instant application since they are being used to perform similar functions to Applicant’s claimed invention”. A wrapper is not a proxy and cannot be substituted for a proxy to

perform a similar function. The two are entirely distinct creatures, operate differently and with different network architecture, to solve different problems.

A wrapper is an interface – it is a layer of software that surrounds the TCP layer. In the present claimed invention, two wrapping layers cooperate to maintain the current state of the TCP connection. A north side wrapper surrounds the application layer and a south side wrapper surround the IP layer. This approach does not affect the software running on the client, does not cause the TCP's implementations to be changed and does not use a proxy.

A proxy is an entirely separate application – it is an agent whereby one system fronts for another. When a proxy is used to mask connection failures, all TCP traffic is redirected between the client and server through the proxy. The proxy maintains the state of connection between the client and the server. If the server crashes, the proxy switches the connection the connection to an alternate server (not cooperate to restart the server, as wrappers of the present invention). A proxy must ensure that sequence numbers of the new connection are consistent with the old connection. This approach introduces a single point of failure – the proxy.

Thus there are significant substantive differences between a wrapper and a proxy that prevent anyone skilled in the art to interpret the proxies taught by Ma to be the wrappers in the present invention.

The Office states that the teachings of Lu demonstrate that wrappers were well known in the art at the time of the present invention, and that the combination of Ma and Lu it would have been obvious to a person skilled in the art to disclose the proxies as wrappers to advantageously provide for means of making server failures transparent to a client while providing layers that would

prevent modification of the connection-oriented protocol. However, Applicant respectfully disagrees with this statement.

There is absolutely no suggestion or motivation to use wrappers as taught by Lu in place of a proxy-server approach taught by Ma for masking server failures. The *wrappers* discussed by Lu are specific to *drivers* – NDIS wrappers. The combination would not result in a functional application as disclosed in the present claimed invention.

NDIS is a network driver interface specification whose purpose is for communication so that a modem device driver doesn't need to communicate directly to a stack driver. (Lu, Col. 17, lines 55-57). Whereas wrappers of the present claimed invention are interposed at a process layer and a network layer, respectively, and are **not** associated with a driver device.

Furthermore, because Lu is directed to DSL technology and a system with a modem device, Lu is not reasonably pertinent to the particular problem with which the present claimed invention was involved. Again, Lu is entirely unrelated to the method of the present claimed invention because it is concerned with communication of multiple modems using DSL. There is no teaching or discussion of recovery of a client-server communication following a crash of the server in Lu. Thus there is no suggestion to modify the teaching of Ma, nor would one skilled in the art be motivated by Lu as it is not related to the problem of client-server recovery following the crash of a server.

Moreover, combining the references of Ma and Lu still fails to teach or suggest all of the claim limitations because the combination fails to teach wrapping layers interposed between the IP layer and the application layer and logging connection state information.

The aforementioned references are devoid of any teaching or suggestion of how to restore a state of connection associated with a connection-oriented layer prior to failure through the use of wrapping layers. There is simply no discussion in either Ma or Lu of restoring connections with a server in a manner as recited in Claim 30. Thus, the cited references do not teach or suggest the method of Claim 30 and for the same reason fail to teach the system of independent Claim 37, either singularly or in combination. Accordingly, there would be no motivation to combine Ma and Lu, to arrive at Claims 30 and 37.

For all the reasons described above, the combination fails to teach or suggest independent Claims 30 or 37.

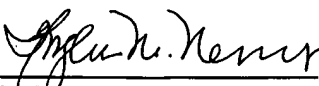
Claims 31-36 depend from Claim 30 and are allowable by virtue of this dependency. Additionally, these claims recite additional features that, when taken together with those of Claim 30, define methods that are not taught or suggested by the Ma and Lu combination.

Conclusion

Pending Claims 30-37 are in condition for allowance. Applicant respectfully requests reconsideration and issuance of the subject application. If any issues remain that preclude issuance of this application, the Examiner is urged to contact the undersigned attorney before issuing a subsequent Action.

Respectfully submitted,
WERNER & AXENFELD, PC

Dated: 30 May 06
PO Box 1629
West Chester, PA 19380
610-701-5810

By 
Kristin M. Nevins
Reg. No. 56,775
Attorney for Applicant